DOCUMENT RESUME

ED 248 862 IR 011 311

AUTHOR	Taitt, Kathy
TITLE	TLC for Growing Minds. APPLE II Programming Manual, Volume 6.
REPORT NO	ISBN-0-88193-006-7
PUB DATE	83
NOTE	71p.; For related documents, see IR 011 306-310, IR 011 314-316, and IR 011 318-319. This manual is also available for TRS-80, IBM PC, Commodore 64, PET, ATARI, TI-99/4A, and VIC-20.
AVAILABLE FROM	Creative Learning Association, Inc., RR#4, Box 330, Charleston, IL 61920 ($9.95 per copy; quantity discounts available).
PUB TYPE	Guides - Classroom Use - Materials (For Learner) (051)
EDRS PRICE	MF01 Plus Postage. PC Not Available from EDRS.
DESCRIPTORS	Adult Education; *Computer Graphics; *Computer Science Education; Computer Software; Input Output Devices; Instructional Materials; *Learning Activities; *Microcomputers; Problem Solving; *Programing; Programing Languages; Secondary Education
IDENTIFIERS	Apple II; *BASIC Programing Language

ABSTRACT

Designed to improve students' thinking, learning, and creative skills while they learn to program a microcomputer in BASIC programing language, this instructional manual for advanced learners provides a brief section of information for each new topic together with a sample program to enter, run on the computer, and observe; a suggested change to clarify the principle being taught, and a challenge requiring use of the new principles. Each chapter includes a review. A chapter on high resolution graphics covers the commands HGR; HCOLOR; HPLOT X, Y; HPLOT X1,Y1 TO X2,Y2; HGR2; and PDL(X). Directions given for making a shape table include making a shape definition and a shape index, putting the shape table into memory, and using the commands CALL-151 and HIMEM. Commands for using shape tables in a program are also explained, i.e., SCALE; ROT; DRAW N AT X,Y; and XDRAW N AT X,Y. Other commands described, including those for storing shape tables on disk, are BSAVE, BLOAD, COPYA, LOCK, and UNLOCK. A reference chart displays commands and their functions, and an appendix lists hexadecimals. Separate color coded project books for junior high and high school/adult students are available for this manual. (Author/LMM)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

THINKING · LEARNING · CREATING

ED248862

U.S. DEPARTMENT OF EDUCATION
NATIONAL INSTITUTE OF EDUCATION
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

✓ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official NIE position or policy.

"PERMISSION TO REPRODUCE THIS MATERIAL IN MICROFICHE ONLY HAS BEEN GRANTED BY

Marilyn Burton

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

IR011311

THINKING · LEARNING · CREATING

TLC

APPLE II®
PROGRAMMING MANUAL

for Growing Minds

Volume 6

by Kathy Taitt

tion Resources

CREATIVE LEARNING ASSOCIATION, INC.
RR#4, Box 330
Charleston, Illinois 61920

ISBN 0-88193-006-7

© 1983 CREATIVE LEARNING ASSOCIATION, INC.

Charleston, Illinois 61920

All rights reserved. No part of this book, except for brief passages in articles and reviews that refer to both author and publisher, may be reproduced without written permission of the publisher.

Table of Contents

Apple Volume VI

CHAPTER 1: HIGH RESOLUTION GRAPHICS Page

HGR 1
HCOLOR 1
HPLOT X,Y 1
HPLOT X1,Y1 TO X2,Y2 4
HGR2 12
PDL(X) 15
Review 18

CHAPTER 2: MAKING A SHAPE TABLE

Making a Shape Definition 20
Making a Shape Index 29
Putting the Shape Table into Memory 31
CALL-151 31
HIMEM 33
Review 35

CHAPTER 3: USING SHAPE TABLES IN YOUR PROGRAM

SCALE 36
ROT 36
DRAW N AT X,Y 36
XDRAW N AT X,Y 40
Review 49

Creative Learning Association
Charleston, IL 61920

Table of Contents (continued)

CHAPTER 4: STORING SHAPE TABLES ON DISK AND OTHER COMMANDS — Page

BSAVE 51
BLOAD 52
COPYA 54
LOCK 57
UNLOCK 57
Review 60
APPENDIX 1 61
Reference 62

CHAPTER 1: HIGH RESOLUTION GRAPHICS

INFORMATION

High-resolution graphics plot much smaller spots on the screen than low-resolution graphics. This means you can create more refined pictures and designs.

The high-resolution commands that correspond to the low-resolution ones you already know are HGR, HCOLOR, and HPLOT.

ENTER/RUN/OBSERVE

```
10 HGR
20 HCOLOR=2
30 HPLOT 0,0
```

If a small dot doesn't appear in the upper left corner of the screen when you RUN this program, change line 20 to HCOLOR=1.

Note: Use TEXT to get out of the high-resolution graphics mode, just as you do in low-resolution graphics.

INFORMATION

The value for HCOLOR can be from Ø to 7. Different television screens produce different colors when given these numbers. To see the colors your set will give, add the lines below.

CHANGE IT/RUN/OBSERVE

```
 5 X=2
15 FOR Z=Ø TO 7
2Ø HCOLOR=Z
3Ø HPLOT X,2
4Ø X=X+5
5Ø NEXT Z
```

Write down on a piece of paper the eight HCOLOR numbers and the colors they produce. You will want to refer to this later.

CHANGE IT/RUN/OBSERVE

Let's compare high-resolution graphics to low-resolution.

```
1Ø GR
2Ø COLOR=Z
3Ø PLOT X,2
```

INFORMATION

Because of the way television screens work, the color of HCOLOR=3 will be blue if the X-coordinate is even, green if it is odd, and white if both (X,Y) and (X+1,Y) are plotted.

ENTER/RUN/OBSERVE

```
10 HGR
20 HCOLOR=3
30 HPLOT 0,0
40 HPLOT 5,0
50 HPLOT 10,0:HPLOT 11,0
```

HCOLOR=7 works the same way, but it produces three different colors.

CHANGE IT/RUN/OBSERVE

```
20 HCOLOR=7
```

CHALLENGE

The values allowed for HPLOT are 0 to 279 for X and 0 to 159 for Y. With this information,

write a program that will place a dot in the four corners of the graphics screen.

ENTER/RUN/OBSERVE

```
10 HGR
20 HCOLOR=2
30 HPLOT 0,0 TO 279,0
```

(Use whatever HCOLOR number is necessary to produce an image on your screen.)

HPLOT X1,Y1 TO X2,Y2 is another form of the HPLOT command.

CHANGE IT/RUN/OBSERVE

```
40 HPLOT 0,0 TO 0,159
```

CHANGE IT/RUN/OBSERVE

```
30 HPLOT 279,0 TO 0,0 TO 0,159 TO 10,159 TO
10,85 TO 200,85

40
```

CHANGE IT/RUN/OBSERVE

HCOLOR can be a little strange. Enter the lines below to see how.

```
 5 HOME
15 FOR C=1 TO 7
20 HCOLOR=C
40 FOR T=1 TO 500:NEXT T
50 NEXT C
70 PRINT "END"
```

CHANGE IT/RUN/OBSERVE

```
60 TEXT
```

Notice the difference line 60 makes.

CHANGE IT/RUN/OBSERVE

```
30 HPLOT 279,0 TO 1,0 TO 1,159 TO 11,159 TO
11,85 TO 200,85
60
70
```

INFORMATION

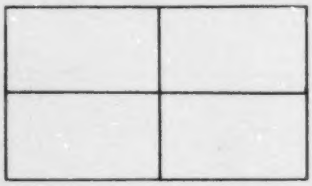
From the last two programs, you can see that HCOLOR=1 and HCOLOR=5 show up only when the X-coordinate is an odd number or when both (X,Y) and (X+1,Y) are plotted.

HCOLOR=2 and HCOLOR=6 appear when the X-coordinate is even or when both (X,Y) and (X+1,Y) are plotted.

(Your HCOLOR numbers may differ from these general guidelines, but a little experimentation should show what you need to know.)

CHALLENGE

Use HPLOT to create this type of pattern on your screen:



ENTER/RUN/OBSERVE

```
10 HGR
20 HCOLOR=2
30 HPLOT X,Y
40 X=X+1:Y=Y+1
50 IF Y=159 THEN END
60 GOTO 30
```

CHANGE IT/RUN/OBSERVE

```
40 X=X+2:Y=Y+1
50 IF X>279 THEN END
```

CHANGE IT/RUN/OBSERVE

```
13 PRINT "YOU WILL DECIDE THE SLOPE OF THE
LINE."
15 INPUT "HOW MUCH DO YOU WANT X TO
INCREASE EACH TIME?";J
18 INPUT "Y?";K
40 X=X+J:Y=Y+K
50 IF X > 279-J THEN END
55 IF Y > 159-K THEN END
```

Creative Learning Association
Charleston, IL 61920

CHANGE IT/RUN/OBSERVE

```
19 INPUT "COLOR? (Ø-7)";C
2Ø HCOLOR=C
```

RUN this program at least five times trying different colors and shapes.

CHANGE IT/RUN/OBSERVE

```
 5 X=Ø:Y=Ø:W=278:Z=Ø
3Ø HPLOT X,Y:HPLOT W,Z
45 W=W-J:Z=Z+K
53 IF W < J THEN END
58 IF Z > 159-K THEN END
```

CHALLENGE

Create a large triangle on the screen using high resolution graphics. Can you make each side of the triangle a different color?

Creative Learning Association
Charleston, IL 61920

ENTER/RUN/OBSERVE

```
 10 HGR
 20 HCOLOR=2:G=0
 30 GOSUB 110
 40 HCOLOR=5:G=11
 50 GOSUB 110
 60 HCOLOR=1:G=22
 70 GOSUB 110
 80 HCOLOR=3:G=33
 90 GOSUB 110
100 END
110 REM *** SUBROUTINE FOR DRAWING BAR ***
120 FOR Y=G TO G+10
130 HPLOT 0,Y TO 279,Y
140 NEXT Y
150 RETURN
```

Note: In lines 20, 40, 60, and 80, use any HCOLOR values that will produce four different colors on your screen.

CHANGE IT/RUN/OBSERVE

```
  5 G=0
 15 FOR N=1 TO 7
 20
```

(program continued)

```
 30
 40 HCOLOR=5
 60
 70
 80 HCOLOR=3
 95 NEXT N
145 G=G+1
```

CHALLENGE

Program the computer to draw a striped pattern on the screen. The stripes can be horizontal, vertical, diagonal, or combinations of all three!

CHANGE IT/RUN/OBSERVE

```
 10 HGR
 20 C=INT(RND(1)*8)
 30 X=INT(RND(1)*280)
 40 Y=INT(RND(1)*160)
 50 HCOLOR=C
 60 HPLOT X,Y
 70 GOTO 20
```

Creative Learning Association
Charleston, IL 61920

CHALLENGE

Make the computer draw a box on the screen and have dots randomly plotted inside the box.

SUPER CHALLENGE

Make the computer draw a box in the center of the screen that uses about half the screen. Have the dots randomly appear anywhere on the screen. MAKE those that are inside the box one color while those outside a different color.

INFORMATION

If your computer has less than 24K of memory, skip to page 15. Otherwise, continue with the program below.

ENTER/RUN/OBSERVE

```
10 HGR
20 HCOLOR=2
30 FOR Y=0 TO 159 STEP 5
40 HPLOT 0,Y TO 279,Y
50 NEXT Y
60 FOR X=0 TO 279 STEP 5
70 HPLOT X,0 TO X,159
80 NEXT X
```

CHANGE IT/RUN/OBSERVE

```
10 HGR2
30 FOR Y=0 TO 191 STEP 5
70 HPLOT X,0 TO X,191
```

Type TEXT (you won't see it on the screen) and press [RETURN] when you want to get out of the graphics mode.

INFORMATION

HGR2 differs from HGR and GR in that it doesn't save the bottom four lines of the screen for TEXT, which increases the amount of space for graphics. In HGR2, the Y-coordinate can be from Ø to 191. The X-coordinate range is still Ø to 279. HGR2 can only be used with computers which have 24K of memory or more.

ENTER/RUN/OBSERVE

Here's an easier way to draw diagonals with HPLOT.

```
10 HGR2
20 HCOLOR=2
30 HPLOT 0,0 TO 279,191
```

CHANGE IT/RUN/OBSERVE

```
40 HPLOT 0,100 TO 279,150
```

CHANGE IT/RUN/OBSERVE

```
30 Y1=INT(RND(1)*192)
40 Y2=INT(RND(1)*192)
50 HPLOT 0,Y1 TO 279,Y2
60 GOTO 30
```

CHALLENGE

A. Create a program that draws lines, randomly choosing the slope of each line.

B. Add commands to your program from part A so that the color of each line is also randomly selected.

Creative Learning Association
Charleston, IL 61920

INFORMATION

If you have game paddles with your computer, the command PDL can be a lot of fun.

PDL(X) returns a number from Ø to 255, depending upon the position of the dial on paddle X. You may have as many as four different game paddles connected at once. Each will have a different value for X. (X can be Ø, 1, 2, or 3.)

ENTER/RUN/OBSERVE

If you have game paddles, you probably have two of them. Hook them up to your computer, then try the program below.

```
10 HOME
20 PRINT PDL(1)
30 GOTO 20
```

This program will display on the screen the value of the paddle setting. Try turning the dial on one of the paddles. See if the numbers change on the screen. Then try the other one. The paddle that causes the numbers to change is paddle #1. The other is paddle #Ø.

CHANGE IT/RUN/OBSERVE

```
20 PRINT PDL(0)
```

ENTER/RUN/OBSERVE

Using PDL(X) is a quick and easy way to input numbers into the computer.

```
10 HOME
20 X=PDL(1)
30 PRINT CHR$(X);" ";
40 IF PDL(1)=X GOTO 30
50 GOTO 20
```

Turn the dial of paddle #1 as this program is running.

ENTER/RUN/OBSERVE

```
10 HGR2
20 HCOLOR=3
30 X=PDL(0)
40 Y=PDL(1)
50 IF Y > 191 THEN Y=191
60 HPLOT X,Y
70 GOTO 30
```

CHALLENGE

Create a program that uses PDL with low-resolution graphics.

REVIEW: Choose the command or key which will cause the function described below.

COMMAND/KEY	FUNCTION
	This command places the computer in the high-resolution graphics mode.
	Use this command to produce colors in high-resolution graphics.
	This command lights one high-resolution square on the screen.
	Use this graphics command when you do not want the TEXT lines available at the bottom of the screen.
	This command, used with game paddles, returns a number between Ø and 255.

CHAPTER 2: MAKING A SHAPE TABLE

INFORMATION

Here's how to make your computer create shapes and save them in memory. Once the shape is there, the computer can draw the shape, enlarge it up to 255 times its original size (SCALE), and rotate the shape through 360° (ROT).

When you put the information the computer needs to create a shape into the computer's memory, that information is called a shape table.

A shape table consists of two parts: one or more shape definitions and a shape index. A shape definition is the instructions for drawing the shape. The shape index is like a table of contents for the shape table. It tells how many shape definitions are in the table and where each one is located in memory.

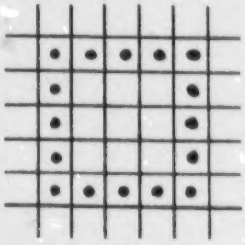
FROM THIS POINT ON, BE SURE YOU HAVE AN INITIALIZED DISK IN THE DRIVE AND THE COMPUTER BOOTED SO YOU CAN SAVE WHAT YOU CREATE.

A. Making a Shape Definition

For the first example, let's create a square.

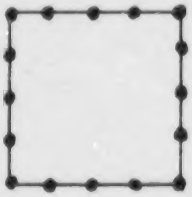
STEP 1

Draw the shape on graph paper by putting dots in the centers of squares. Our square will have five dots on a side.



STEP 2

Next, connect the dots using only straight lines and right (90°) angles.



STEP 3

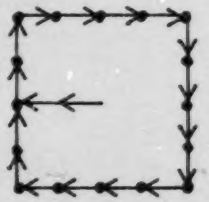
Now you are going to change the squares into a series of little arrows. (These arrows are called vectors.) There are two kinds of arrows:

↑ and ↑• . The first one, ↑ , tells the computer to move. The second, ↑• , tells the computer to plot a point, and then move. Vectors can go in four directions: ↑ , → , ↓ , and ← .

So, all together, there are eight possible types of vectors: ↑ ← ↓ → ↑• ←• ↓• →•

Start drawing in the center of the shape so that if the computer is drawing several of the shapes at the same time in different sizes, they will all have the same center; your design will be symmetrical. Later we'll create a shape definition that doesn't start at its center.

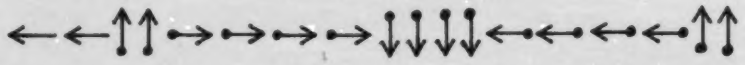
Each vector must be exactly one square long.



Creative Learning Association
Charleston, IL 61920

STEP 4

Mentally take the vectors off the paper and put them in a straight line. Follow the same order that you have on graph paper.



STEP 5

You are now going to arrange the vectors in a table like the one below. This table represents the way that the vectors are stored in the computer's memory.

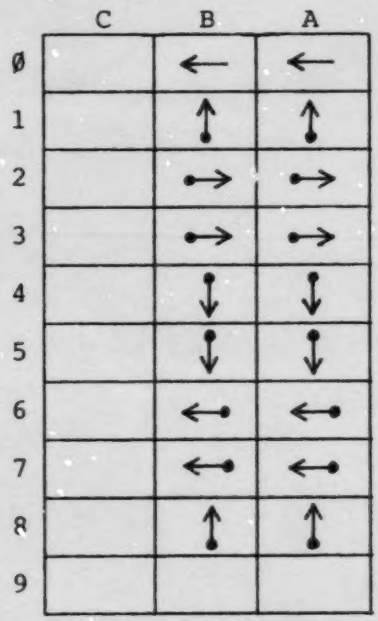
	C	B	A
Ø			
1			
2			
3			
4			
5			
6			
7			
8			
9			

Creative Learning Association
Charleston, IL 61920

The numbers on the left tell the byte number. Each row is one byte of memory.

The computer reads the table from right to left, so the first vector goes in Column A, row Ø. The next goes in Column B, row Ø, and so on.

Notice how the vectors are arranged in the table, then read on to discover the special rules for Column C.



Creative Learning Association
Charleston, IL 61920

Column C is not allowed to contain any vectors that plot or ones that move up. The only vectors that are allowed in Column C are →, ←, and ↓. It just so happens that this first example only contains two of these vectors, and they are in Columns A and B.

The remaining step in constructing a shape definition is coding the vectors into numbers the computer can understand. To do this, you need to become familiar with two number systems: binary and hexadecimal.

Decimal Number	=	Binary Number	=	Hexadecimal Number
Ø		ØØØØ		Ø
1		ØØØ1		1
2		ØØ1Ø		2
3		ØØ11		3
4		Ø1ØØ		4
5		Ø1Ø1		5
6		Ø11Ø		6
7		Ø111		7
8		1ØØØ		8
9		1ØØ1		9
1Ø		1Ø1Ø		A
11		1Ø11		B
12		11ØØ		C
13		11Ø1		D
14		111Ø		E
15		1111		F

The system you are most familiar with, the decimal number system, has ten symbols (Ø, 1, 2, 3, 4, 5, 6, 7, 8, 9). The binary system has two symbols (Ø and 1), and the hexadecimal system has sixteen symbols (Ø, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F).

In the Appendix is a more complete chart for converting decimal numbers to hexadecimal.

Now, back to our shape definition.

Each of the eight vectors has an unique binary code number. The first digit of the binary number tells whether or not a point is being plotted (Ø = do not plot, 1 = plot). The remaining two digits indicate the direction to move.

Vector	Binary Number
	ØØØ
	ØØ1
	Ø1Ø
	Ø11
	1ØØ
	1Ø1
	11Ø
	111

STEP 6

Code each vector in the shape table from Step 5 to its corresponding binary code number. Fill in all the blanks with Ø's. Always finish the shape definition with a complete row of zeros. This indicates to the computer that this is the end of the shape definition.

Note: Since Column C only contains vectors that do not plot, the first binary number in this column will always be zero. For this reason, the first digit (Ø) is omitted, and Column C contains only two digits.

	C	B	A
Ø	ØØ	←	←
1	ØØ	↑	↑
2	ØØ	→	→
3	ØØ	→	→
4	ØØ	↓	↓
5	ØØ	↓	↓
6	ØØ	←	←
7	ØØ	←	←
8	ØØ	↑	↑
9	ØØ		

plus

Vector	Binary Number
↑	ØØØ
→	ØØ1
↓	Ø1Ø
←	Ø11
↑	1ØØ
→	1Ø1
↓	11Ø
←	111

plus

produces

	C	B	A
Ø	ØØ	Ø11	Ø11
1	ØØ	1ØØ	1ØØ
2	ØØ	1Ø1	1Ø1
3	ØØ	1Ø1	1Ø1
4	ØØ	11Ø	11Ø
5	ØØ	11Ø	11Ø
6	ØØ	111	111
7	ØØ	111	111
8	ØØ	1ØØ	1ØØ
9	ØØ	ØØØ	ØØØ

which can be grouped into two groups of four digits

	C	B A
Ø	ØØØ1	1Ø11
1	ØØ1Ø	Ø1ØØ
2	ØØ1Ø	11Ø1
3	ØØ1Ø	11Ø1
4	ØØ11	Ø11Ø
5	ØØ11	Ø11Ø
6	ØØ11	1111
7	ØØ11	1111
8	ØØ1Ø	Ø1ØØ
9	ØØØØ	ØØØØ

STEP 7

To enter the shape table into the computer, the binary numbers must be in hexadecimal form. Now convert the binary codes to hexadecimal.

Binary Number	Hexadecimal Number
ØØØØ	Ø
ØØØ1	1
ØØ1Ø	2
ØØ11	3
Ø1ØØ	4
Ø1Ø1	5
Ø11Ø	6
Ø111	7
1ØØØ	8
1ØØ1	9
1Ø1Ø	A
1Ø11	B
11ØØ	C
11Ø1	D
111Ø	E
1111	F

	C	B	A		Hexadecimal Number
Ø	ØØØ1		1Ø11	=	1B
1	ØØ1Ø		Ø1ØØ	=	24
2	ØØ1Ø		11Ø1	=	2D
3	ØØ1Ø		11Ø1	=	2D
4	ØØ11		Ø11Ø	=	36
5	ØØ11		Ø11Ø	=	36
6	ØØ11		1111	=	3F
7	ØØ11		1111	=	3F
8	ØØ1Ø		Ø1ØØ	=	24
9	ØØØØ		ØØØØ	=	ØØ
	First Hex Digit		Second Hex Digit		

You have finished the shape definition! The shape index will just take a minute.

B. Making a Shape Index

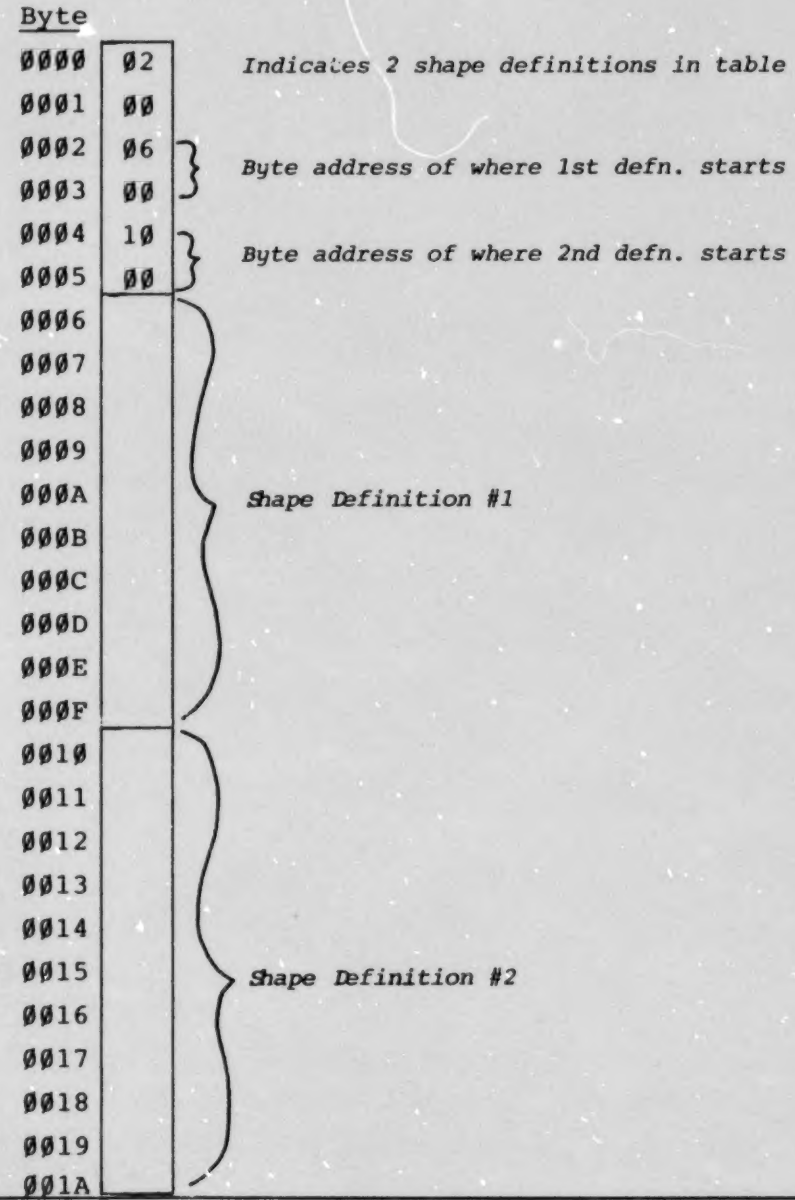
Think of the shape index and the shape definition as two columns. When the shape index is placed on top of the shape definition, the result is a shape table. Each row in the shape table holds two digits of information.

Shape Index

The first row (Ø) of the shape index tells the number of shape definitions in the table. In this case, this value is one. The second row (1) is always ØØ. Rows ØØØ2 and ØØØ3 give the byte number (in hexadecimal) of where the first shape definition starts in the table. Our shape definition starts at byte ØØØ4. As before, the computer reads bytes from right to left. So the right-most two digits of ØØØ4 go in Row ØØØ2 and the left two digits of ØØØ4 go in Row ØØØ3.

MEMORY LOCATIONS	SHAPE TABLE	
ØØØØ	Ø 1	SHAPE INDEX
ØØØ1	Ø Ø	
ØØØ2	Ø 4	
ØØØ3	Ø Ø	
ØØØ4	1 B	SHAPE DEFINITION
ØØØ5	2 4	
ØØØ6	2 D	
ØØØ7	2 D	
ØØØ8	3 6	
ØØØ9	3 6	
ØØØA	3 F	
ØØØB	3 F	
ØØØC	2 4	
ØØØD	Ø Ø	

Suppose your shape table had two shape definitions. The shape index would look like this:



Creative Learning Association
Charleston, IL 61920

C. Putting the Shape Table into the Computer's Memory

To put the shape table into the computer's memory, the computer must be in the monitor mode.

To get into the monitor mode, type

```
**************************************************
*                                                *
* CALL-151                                       *
*                                                *
**************************************************
```

and press RETURN . You can tell that the computer is in the monitor mode when the cursor symbol (]) is replaced by an asterisk (*).

(The command CALL causes the computer to follow a subroutine stored at a certain memory location. The memory location must be in the range -65535 through 65535. To put the computer into the monitor mode, we need to call the subroutine at location -151.)

To put the shape table into the computer, you must choose a location in memory to place it. A good location for the first byte of the table is 1DFC. (1DFC is a hexadecimal number which equals the decimal number 7676.)

This location is not in the memory area used by HGR or HGR2 and should be out of the way of other program operations.

In the monitor mode, type

```
1DFC
```

and press RETURN .

The number displayed on the screen is what is currently stored in this location.

To enter your shape table, you will type 1DFC: followed by the bytes of the shape table separated by spaces.

Enter

```
1DFC:Ø1 ØØ Ø4 ØØ 1B 24 2D 2D 36 36 3F 3F 24
ØØ
```

To see that the table is stored correctly, enter

```
1DFC
```

and press the RETURN key four times.

INFORMATION

The command HIMEM: sets the highest memory location available for BASIC programs. To ensure that your shape table won't be erased by a program, set HIMEM (Hi Memory) to 1DFC, the start of the shape table.

While still in the monitor mode, enter

```
**************************************************
*                                                *
* 73:FC 1D                                       *
*                                                *
**************************************************
```

Notice that, as before, the right-most two digits (FC) come first.

Be sure to hit the number one instead of the letter I!! If you hit I, the computer goes into reverse.

73 is the hexadecimal location that corresponds to the HIMEM command. If you were in BASIC, you would enter

```
**************************************************
*                                                *
* HIMEM:7676                                     *
*                                                *
**************************************************
```

The shape table is now stored in memory starting at hexadecimal location 1DFC. When you run a program, however, and the computer encounters a command to DRAW a shape, it looks in hex memory location E8 to find out where the shape table is stored. So, you need to put the value 1DFC into location E8.

Type

```
E8:FC 1D
```

and press [RETURN] .

Believe it or not, your shape table is now safely tucked away in memory and you're ready to begin programming. To get back into BASIC, press [CTRL] [C] and then [RETURN] .

You'll create more shapes later!

Creative Learning Association
Charleston, IL 61920

REVIEW: Choose the command or key which will cause the function described below.

COMMAND/KEY	FUNCTION
	This is the term for information the computer needs to create a shape consisting of one or more shape definitions and a shape index.
	These are the instructions for drawing the shape.
	This tells how many shape definitions are in the table and where each one is located in memory.
	Use this command to put the computer into the monitor mode.
	This command sets the highest memory location available for BASIC programs.

Creative Learning Association
Charleston, IL 61920

CHAPTER 3: USING SHAPE TABLES IN YOUR PROGRAM

INFORMATION

The command SCALE determines the size of the shape drawn. The value for SCALE can be 1 to 255, with 1 being the original size of the shape.

The command ROT allows the shape to be rotated from its original position. The values for ROT range from Ø to 255.

DRAW, as you can guess, is the command which actually draws the shape on the screen. The format for DRAW is

DRAW N AT X,Y

where N is the number of the shape definition, and X,Y is the point on the screen where the drawing starts. In the first example, the first vector, ← , starts from the center of the screen, at point 139,79.

The commands HCOLOR, SCALE, and ROT must always precede DRAW.

ENTER/RUN/OBSERVE

```
10 HGR
20 HCOLOR=1
30 SCALE=1
40 ROT=1
50 DRAW 1 AT 139,79
```

CHANGE IT/RUN/OBSERVE

```
50 DRAW 1 AT 5,100
60 DRAW 1 AT 205,40
```

CHANGE IT/RUN/OBSERVE

```
30 SCALE=5
50
```

CHANGE IT/RUN/OBSERVE

```
25 FOR S=1 TO 255
30 SCALE=S
60 DRAW 1 AT 139,79
70 PRINT "S=";S
80 NEXT S
```

At what value of S is the screen completely filled?

CHANGE IT/RUN/OBSERVE

```
20
35 HCOLOR=C
75 C=C+1:IF C=8 THEN C=0
```

CHANGE IT/RUN/OBSERVE

```
25 FOR S=1 TO 75
40
28 ROT=S
```

ENTER/RUN/OBSERVE

```
10 HGR
20 HCOLOR=3
30 SCALE=25
40 FOR R=0 TO 16
50 ROT=R
60 DRAW 1 AT 139,79
70 FOR T=1 TO 255:NEXT T
80 NEXT R
```

CHALLENGE

Change the last program so that the design will be printed in all eight colors.

Creative Learning Association
Charleston, IL 61920

INFORMATION

It has already been said that the command ROT can have values of Ø to 255. RUN the program below and see if you can determine which ROT values correspond to 45°, 90°, 135°, 18Ø°, 225°, 27Ø°, and 315°.

CHANGE IT/RUN/OBSERVE

```
1Ø HGR
2Ø HCOLOR=3
3Ø SCALE=25
4Ø FOR R=Ø TO 75
5Ø ROT=R
6Ø DRAW 1 AT 139,79
7Ø PRINT "R=";R
8Ø FOR T=1 TO 25Ø:NEXT T
9Ø XDRAW 1 AT 139,79
1ØØ NEXT R
```

Creative Learning Association
Charleston, IL 61920

INFORMATION

As you probably figured out, every eight ROT numbers equal 45°. So ROT 16 = 90° rotation, ROT 32 = 180°, and so on.

The command XDRAW draws the shape in the complement of the color the shape was DRAWn in. This is an easy way to erase. The complement of white is black, the complement of blue is green, and vice versa. XDRAW has the same format as DRAW.

CHANGE IT/RUN/OBSERVE

```
40 ROT=0
50
70
100
```

CHANGE IT/RUN/OBSERVE

```
100 GOTO 80
```

The complement of an XDRAW is the same color as DRAW.

CHANGE IT/RUN/OBSERVE

```
**************************************************
*                                                *
*  2Ø HCOLOR=1                                   *
*                                                *
**************************************************
```

CHANGE IT/RUN/OBSERVE

```
**************************************************
*                                                *
*  2Ø HCOLOR=2                                   *
*                                                *
**************************************************
```

CHANGE IT/RUN/OBSERVE

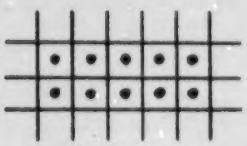
```
**************************************************
*                                                *
*  2Ø HCOLOR=4                                   *
*                                                *
**************************************************
```

Try the remaining three color codes to see what their complements are.

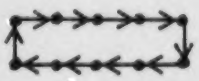
INFORMATION

Let's do another shape. This time we will make a rectangle. I'll go through the steps rather quickly since you already know what to do.

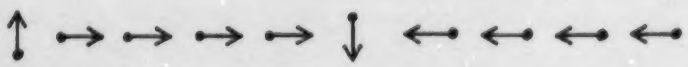
STEP 1



STEPS 2 & 3



STEP 4



Notice that this shape definition starts in the corner of the shape, not the center.

STEP 5

C	B	A

STEP 6

	C	B	A
Ø	ØØ	1Ø1	1ØØ
1	ØØ	1Ø1	1Ø1
2	ØØ	11Ø	1Ø1
3	ØØ	111	111
4	ØØ	111	111
5	ØØ	ØØØ	ØØØ

Creative Learning Association
Charleston, IL 61920

STEP 7

Convert bytes Ø to 5 from binary into hexadecimal notation.

	C	B	A	
Ø	ØØ	1Ø1	1ØØ	=
1	ØØ	1Ø1	1Ø1	=
2	ØØ	11Ø	1Ø1	=
3	ØØ	111	111	=
4	ØØ	111	111	=
5	ØØ	ØØØ	ØØØ	=

Your solution should be

2C 2D 35 3F 3F ØØ

but don't accept it until you agree!

ERIC Clearinghouse on Information Resources
School of Education
Syracuse University
Syracuse, New York 13210
(315) 423-3640

We are now going to make a shape table with both the rectangle and the square definition.

Byte #		
ØØØØ	Ø2	*The shape table has 2 shape defns.*
ØØØ1	ØØ	
ØØØ2	Ø6	*Starting address of Definition #1.*
ØØØ3	ØØ	
ØØØ4	ØC	*Starting address of Definition #2.*
ØØØ5	ØØ	
ØØØ6	2C	*Definition #1, The rectangle*
ØØØ7	2D	
ØØØ8	35	
ØØØ9	3F	
ØØØA	3F	
ØØØB	ØØ	
ØØØC	1B	*Definition #2, The square*
ØØØD	24	
ØØØE	2D	
ØØØF	2D	
ØØ1Ø	36	
ØØ11	36	
ØØ12	3F	
ØØ13	3F	
ØØ14	24	
ØØ15	ØØ	

Creative Learning Association
Charleston, IL 61920

ENTER/RUN/OBSERVE

Enter the monitor mode

```
CALL -151
```

Now enter

```
1DFC:  Ø2 ØØ Ø6 ØØ ØC ØØ 2C 2D 35 3F 3F ØØ
1B 24 2D 2D 36 36 3F 3F 24 ØØ
```

Then enter

```
E8:FC 1D
```

and

```
73:FC 1D
```

To return to BASIC, press CTRL C and then RETURN .

ENTER/RUN/OBSERVE

```
1Ø HGR
2Ø SCALE=2Ø
3Ø FOR R=Ø TO 64
4Ø HCOLOR=3
```

(program continued)

```
50 ROT=R
60 DRAW 1 AT 139,79
70 FOR T=1 TO 250:NEXT T
80 XDRAW 1 AT 139,79
90 NEXT R
```

CHANGE IT/RUN/OBSERVE

```
65 DRAW 2 AT 139,79
85 XDRAW 2 AT 139,79
```

Notice the stray dots!

CHANGE IT/RUN/OBSERVE

```
75 HCOLOR=0
80 DRAW 1 AT 139,79
85 DRAW 2 AT 139,79
```

CHANGE IT/RUN/OBSERVE

```
30 FOR R=1 TO 125
60 DRAW 1 AT X,79
70 X=X+2:IF X > 279 THEN X=0
```

CHANGE IT/RUN/OBSERVE

```
 30 FOR R=1 TO 255
100 GOTO 30
```

ENTER/RUN/OBSERVE

```
 10 HGR2
 20 SCALE=5
 30 C=INT(RND(1)*8)
 40 X=INT(RND(1)*280)
 50 Y=INT(RND(1)*192)
 60 FOR R=1 TO 64 STEP 4
 70 ROT=R
 80 HCOLOR=C
 90 DRAW 1 AT X,Y
100 NEXT R
110 GOTO 30
```

CHANGE IT/RUN/OBSERVE

```
 20
 60 Z=INT(RND(1)*3)
 63 IF Z=1 THEN SCALE=5
 65 IF Z=2 THEN SCALE=10
 70 ROT=0
 90 DRAW Z AT X,Y
100
```

CHANGE IT/RUN/OBSERVE

```
 20 IF Z=2 THEN XDRAW 2 AT X,Y
 30 HCOLOR=3
 80
110 GOTO 20
```

CHALLENGE

Write a program that makes the square rotate on one side of the screen and the rectangle rotate on the other side.

CHALLENGE

By now you should realize that there are many, many types of designs you can create with high-resolution shape programs.

Use your imagination, be creative, and experiment with the commands ROT, SCALE, HCOLOR, DRAW, and XDRAW to create an interesting program.

Don't turn your computer off without learning how to save the shape tables on disk. The next chapter shows you how.

REVIEW: Choose the command or key which will cause the function described below.

COMMAND/KEY	FUNCTION
	This command determines the size of the shape drawn and can hold values between 1 and 255.
	Use this command to rotate the shape from its original position. Its values range from Ø to 255.
	This command draws the shape on the screen at location X,Y.
	This command draws the shape in the complement of the color used with the DRAW command.

CHAPTER 4: STORING SHAPE TABLES ON DISK AND OTHER DISK COMMANDS

INFORMATION

After you have taken the time to make a shape table and put it into the computer's memory, you would probably like to save it on disk. Otherwise, it will be lost when the computer is turned off.

To save a shape table on disk, you must first know the length of the shape table. Each pair of the shape table's numbers is stored in one byte of memory.

Therefore, our shape table (Ø2 ØØ Ø6 ØØ ØC ØØ 2C 2D 35 3F 3F ØØ 1B 24 2D 2D 36 36 3F 3F 24 ØØ) is 22 bytes long.

INFORMATION

The command for saving shape tables on disk is BSAVE. The format for BSAVE is

BSAVE *name of shape table*, A$ *starting address of shape table*, L *length of the table in decimal.*

ENTER/RUN/OBSERVE

With the computer booted, an initialized disk in the drive, and the shape table in memory, enter

```
**********************************************
*                                            *
* BSAVE RECTANGLE/SQUARE,A$ 1DFC,L22         *
*                                            *
**********************************************
```

The name of the shape table follows all the normal rules for filenames.

ENTER/RUN/OBSERVE

Enter

```
**********************************************
*                                            *
* CATALOG                                    *
*                                            *
**********************************************
```

to be sure the shape table has been stored on disk.

Loading Shape Tables from Disk

To load a shape table from disk into the computer's memory, enter

```
BLOAD RECTANGLE/SQUARE
```

BLOAD puts the shape table back into the memory address it was BSAVEd from. In this case, this is location 1DFC.

BLOAD doesn't automatically put the starting address into location #8; you have to do it yourself.

Now enter

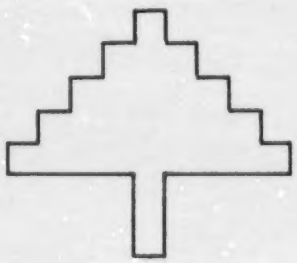
```
CALL -151
E8:FC 1D
```

That's all there is to it!

Remember that the number 1DFC must be entered in reverse order, FC 1D.

CHALLENGE

A. Construct a shape table with this shape. Save it on a disk.



B. Create a shape table for a shape that represents an animal. Save it on a disk.

C. Now create a program that uses your two shapes.

INFORMATION

COPYA is a program on the System Master diskette that enables you to make copies of disks.

You can use COPYA whether you have one or two disk drives. The disk that is going to be copied onto doesn't need to be initialized beforehand. Any information that is already on the disk will be erased.

Using COPYA with one disk drive

Put the System Master disk in the drive. Boot the disk by entering: PR#6.

Enter

```
**********************************************
*                                            *
* RUN COPYA                                  *
*                                            *
**********************************************
```

The computer assumes you have the card for the drive inserted in slot #6 inside the computer. "ORIGINAL SLOT:DEFAULT=6" will appear on the screen. If your card is not in slot #6, type the number of the slot the card is in and press RETURN . Your disk is in drive #1, so just press RETURN

Creative Learning Association
Charleston, IL 61920

when "ORIGINAL DRIVE:DEFAULT=1" appears. Again, if your card is in slot #6, just press [RETURN] for "DUPLICATE SLOT:DEFAULT=6". Otherwise, enter the correct slot number. When "DUPLICATE DRIVE:DEFAULT=2" appears, type [1] and press [RETURN] .

When the message "PRESS 'RETURN' KEY TO BEGIN COPY" appears, remove the System Master disk and insert the disk you want to make a copy of into the drive.

From this point on, just follow the instructions given by the COPYA program. "ORIGINAL" refers to the disk you want to make a copy of, and "DUPLICATE" refers to the blank disk. Many times throughout the program you will be asked to take out the original disk, insert the duplicate, and vice versa.

Using COPYA with two disk drives

Put the System Master disk in drive #1 and boot it by entering PR#6. Insert the blank disk into drive #2.

Enter

```
**************************************************
*                                                *
*  RUN COPYA                                     *
*                                                *
**************************************************
```

The COPYA program (on disk in drive #1) takes it from here. Follow the instructions, and as long as your values coincide with the default values, just press the [RETURN] key. If the disk drive card is in a slot other than #6 in the computer, when "SLOT:DEFAULT=6" comes on the screen, type the number of the correct slot and press [RETURN] .

When "PRESS 'RETURN' KEY TO BEGIN COPY" appears, remove the System Master disk from drive #1 and replace it with the disk you want to make a copy of. From here on, just follow the instructions given by the COPYA program. "ORIGINAL" refers to the disk you want to make a copy of, and "DUPLICATE" refers to the blank disk.

INFORMATION

The disk command LOCK protects programs on disk from being erased.

When you enter

```
LOCK filename, drive number
```

the program with that filename in the drive specified can't be erased until you UNLOCK it. An asterisk (*) appears next to the filename of a program that is LOCKed in the CATALOG of a disk.

The format for UNLOCK is the same as for LOCK:

UNLOCK *filename, drive number*

(When the drive number = 1, it doesn't need to be specified in either command.)

ENTER/RUN/OBSERVE

(Have a disk in the drive for this exercise.)

```
10 HOME
20 PRINT "HELLO your name!"
30 GOTO 30
```

ENTER/RUN/OBSERVE

```
**************************************************
*                                                *
* SAVE NAME                                      *
*                                                *
**************************************************
```

ENTER/RUN/OBSERVE

```
**************************************************
*                                                *
* CATALOG                                        *
*                                                *
**************************************************
```

ENTER/RUN/OBSERVE

```
**************************************************
*                                                *
* LOCK NAME                                      *
*                                                *
**************************************************
```

Now ask for a CATALOG. Notice the asterisk (*)?

ENTER/RUN/OBSERVE

```
**************************************************
*                                                *
* DELETE NAME                                    *
*                                                *
**************************************************
```

What happens when you try to execute this command?

ENTER/RUN/OBSERVE

```
**************************************************
*                                                *
* UNLOCK NAME                                    *
*                                                *
**************************************************
```

ENTER/RUN/OBSERVE

```
**************************************************
*                                                *
* CATALOG                                        *
*                                                *
**************************************************
```

ENTER/RUN/OBSERVE

```
**************************************************
*                                                *
* DELETE NAME                                    *
*                                                *
**************************************************
```

ENTER/RUN/OBSERVE

```
**************************************************
*                                                *
* CATALOG                                        *
*                                                *
**************************************************
```

REVIEW: Choose the command or key which will cause the function described below.

COMMAND/KEY	FUNCTION
	Use this command to save shape tables on disk.
	To load a shape table from disk into the computer's memory, use this command.
	This program on the System Master diskette enables you to make copies of disks.
	Use this command to protect programs on disk from being erased.
	This command removes the protection of the LOCK command.

Creative Learning Association
Charleston, IL 61920

APPENDIX 1

Decimal	Hex	Decimal	Hex	Decimal	Hex
1	ØØØ1	27	ØØ1B	53	ØØ35
2	ØØØ2	28	ØØ1C	54	ØØ36
3	ØØØ3	29	ØØ1D	55	ØØ37
4	ØØØ4	3Ø	ØØ1E	56	ØØ38
5	ØØØ5	31	ØØ1F	57	ØØ39
6	ØØØ6	32	ØØ2Ø	58	ØØ3A
7	ØØØ7	33	ØØ21	59	ØØ3B
8	ØØØ8	34	ØØ22	6Ø	ØØ3C
9	ØØØ9	35	ØØ23	61	ØØ3D
1Ø	ØØØA	36	ØØ24	62	ØØ3E
11	ØØØB	37	ØØ25	63	ØØ3F
12	ØØØC	38	ØØ26	64	ØØ4Ø
13	ØØØD	39	ØØ27	65	ØØ41
14	ØØØE	4Ø	ØØ28	66	ØØ42
15	ØØØF	41	ØØ29	67	ØØ43
16	ØØ1Ø	42	ØØ2A	68	ØØ44
17	ØØ11	43	ØØ2B	69	ØØ45
18	ØØ12	44	ØØ2C	7Ø	ØØ46
19	ØØ13	45	ØØ2D	71	ØØ47
2Ø	ØØ14	46	ØØ2E	72	ØØ48
21	ØØ15	47	ØØ2F	73	ØØ49
22	ØØ16	48	ØØ3Ø	74	ØØ4A
23	ØØ17	49	ØØ31	75	ØØ4B
24	ØØ18	5Ø	ØØ32	76	ØØ4C
25	ØØ19	51	ØØ33	77	ØØ4D
26	ØØ1A	52	ØØ34	78	ØØ4E

REFERENCE

COMMAND/KEY	FUNCTION
BLOAD	To load a shape table from disk into the computer's memory, use this command.
BSAVE	Use this command to save shape tables on disk.
CALL-151	Use this command to put the computer into the monitor mode.
COPYA	This program on the System Master diskette enables you to make copies of disks.
DRAW N AT X,Y	This command draws the shape on the screen at location X,Y.
HCOLOR	Use this command to produce colors in high-resolution graphics.
HGR	This command places the computer in the high-resolution graphics mode.
HGR2	Use this graphics command when you do not want the TEXT lines available at the bottom of the screen.

REFERENCE (continued)

COMMAND/KEY	FUNCTION
HIMEM	This command sets the highest memory location available for BASIC commands.
HPLOT X,Y	This command lights one high-resolution square on the screen.
LOCK	Use this command to protect programs on disk from being erased.
PDL(X)	This command, used with game paddles, returns a number between Ø and 255.
ROT	Use this command to rotate the shape from its original position. Its values range from Ø to 255.
SCALE	This command determines the size of the shape drawn and can hold values between 1 and 255.
SHAPE DEFINITION	These are the instructions for drawing the shape.
SHAPE INDEX	This tells how many shape definitions are in the table and where each one is located in memory.

REFERENCE (continued)

COMMAND/KEY	FUNCTION
SHAPE TABLE	This is the term for information the computer needs to create a shape consisting of one or more shape definitions and a shape index.
UNLOCK	This command removes the protection of the LOCK command.
XDRAW N AT X,Y	This command draws the shape in the complement of the color used with the DRAW command.

Creative Learning Association
Charleston, IL 61920

CONGRATULATIONS!!!

You have now completed Volume 6. The INDIGO pages of the ADVANCED PROJECT manual are next. They will give you a chance to learn new ways to use the commands you have already learned.

There is also a Volume 7 for the APPLE series which is an Introduction to Machine Language programming. It assumes you have done the ADVANCED PROJECTS and can use hexadecimal notation.

You are now an excellent programmer. Continue to use your skills to solve problems of interest to you. Keep your skills sharply honed, and they will serve you well in the future.

Now on to the INDIGO PROJECTS!

Creative Learning Association
Charleston, IL 61920